

## File Merge

For this task, there are 2 files of data. Each file contains student data: student ID and grade, each student for each line. Data in the file is ordered by faculty ID (the last 2 digits of student ID). For the same faculty, the data is ordered by student ID.

data1.txt	data2.txt
5830548121 2.50	5930558121 2.30
6031087221 3.12	6231082221 2.12
6130351221 3.20	6030532324 3.87
6230432722 2.45	6030121526 2.99
6230550322 3.23	
6130518324 3.78	
6230215224 2.10	

Your task is implementing a program which can read data from 2 files and combine them to display ordered by faculty ID and student ID. See the example from data1.txt and data2.txt below.

```
5830548121 2.50
5930558121 2.30
6031087221 3.12
6130351221 3.20
6231082221 2.12
6230432722 2.45
6230550322 3.23
6030532324 3.87
6130518324 3.78
6230215224 2.10
6030121526 2.99
```

For practicing, you should implement this program without using list.

(except “split” command)

## Input

Name of both files on the same line, separated by a space.

## Output

List of data combined from 2 files ordering like in the example above.

## Example

Input (from keyboard)		Output (on screen)				
data1.txt data2.txt		5831111121 2.50 6032222221 3.12 6133333321 3.20 6231111122 2.45 6232222222 3.23				
<table><tr><td>data1.txt</td><td>data2.txt</td></tr><tr><td>5831111121 2.50 6032222221 3.12 6133333321 3.20 6231111122 2.45 6232222222 3.23</td><td></td></tr></table>	data1.txt	data2.txt	5831111121 2.50 6032222221 3.12 6133333321 3.20 6231111122 2.45 6232222222 3.23			
data1.txt	data2.txt					
5831111121 2.50 6032222221 3.12 6133333321 3.20 6231111122 2.45 6232222222 3.23						
data3.txt data4.txt		5931111121 2.66 6132222221 2.12 6231111122 2.13 5841111126 2.77 6042222226 2.44 6141111128 3.20 6232222228 3.99				
<table><tr><td>data3.txt</td><td>data4.txt</td></tr><tr><td>5841111126 2.77 6042222226 2.44 6141111128 3.20 6232222228 3.99</td><td>5931111121 2.66 6132222221 2.12 6231111122 2.13</td></tr></table>	data3.txt	data4.txt	5841111126 2.77 6042222226 2.44 6141111128 3.20 6232222228 3.99	5931111121 2.66 6132222221 2.12 6231111122 2.13		
data3.txt	data4.txt					
5841111126 2.77 6042222226 2.44 6141111128 3.20 6232222228 3.99	5931111121 2.66 6132222221 2.12 6231111122 2.13					
data3.txt data4.txt		5841111121 2.77 5931111121 2.66 6042222221 2.44 6132222221 2.12 6231111122 2.13 6141111128 3.20 6232222228 3.99				
<table><tr><td>data3.txt</td><td>data4.txt</td></tr><tr><td>5841111121 2.77 6042222221 2.44 6141111128 3.20 6232222228 3.99</td><td>5931111121 2.66 6132222221 2.12 6231111122 2.13</td></tr></table>	data3.txt	data4.txt	5841111121 2.77 6042222221 2.44 6141111128 3.20 6232222228 3.99	5931111121 2.66 6132222221 2.12 6231111122 2.13		
data3.txt	data4.txt					
5841111121 2.77 6042222221 2.44 6141111128 3.20 6232222228 3.99	5931111121 2.66 6132222221 2.12 6231111122 2.13					

## Hint

To read these files, you may have to use readline command to read data line by line.

Command readline always return something; If there is any data, it will return that string data. On the other hand, if there is no data, it will return the blank string, which is 0 length, instead.

For this task, you have to read data from the file for many times. We suggest you to use the function `read_next` on the next page.

```
def read_next(f):  
    while True:  
        t = f.readline()  
        if len(t) == 0:          # End the loop when all lines are read  
            break  
        x = t.strip().split()    # Remove blanks  
        if len(x) == 2:         # Split input to 2 -> Return the value  
            return x[0], x[1]  
    return "", ""               # When there is no file, return blank string
```

dd